Teoría de Lenguajes - Segundo Parcial

Segundo cuatrimestre de 2021

No usar celulares antes de subir los ejercicios resueltos.
Hacer cada ejercicio en hojas separadas.
Poner nombre, número de libreta y firma en cada página.
Justificar todas las respuestas.
El examen es a libro abierto.
Se aprueba con al menos 65 puntos.

1. (34 pts) Dada la siguiente gramática: $G_1 =< \{E, A, B\}, \{p, \wedge, \neg, (,)\} >$, con P:

$$\begin{aligned} &E \to A \mid \neg E \mid E \ \wedge \ E \mid (E) \\ &A \to p \mid pB \\ &B \to AB \end{aligned}$$

   a) En caso en que los haya, eliminar los símbolos inútiles.

   b) Dar una gramática extendida que genere $L(G_1)$ y sea ELL(1).

   c) Dar su parser iterativo recursivo.

2. (33 pts) Dada la siguiente gramática $G_2 =< \{E, A\}, \{p, \wedge, (,)\} >$, con P:

$$\begin{aligned} &E \to A \mid E \ \wedge \ E \mid (E) \\ &A \to p \end{aligned}$$

   a) Determinar si G_2 es LR(1). Si no lo es, discutir si se pueden resolver los conflictos de la tabla de modo que se preserve $L(G_2)$.

   b) Determinar si G_2 es LR(0).

3. (33 pts) Dada la siguiente gramática: $G_3 =< \{S, B, N\}, \{(,),]\}, P, S >$, con P:

$$\begin{aligned} &S \to S(B) \mid S(N] \mid \lambda \\ &B \to B(B) \mid \lambda \\ &N \to (B)N \mid (N \mid \lambda \end{aligned}$$

   Agregar a G_3 atributos y/o reglas semánticas de modo que resulte una gramática de atributos que permita sintetizar en el atributo *exp* de S la expresión reconocida pero en la cual se ha reemplazado cada corchete por la cantidad de paréntesis de cierre necesaria para que los paréntesis queden balanceados. Por ejemplo, para la cadena de entrada (((](()((]((el resultado debe ser (())(()(()))().

Comentario ejercicio 1: Todo ok. (34/34)

HOJA N°

FECHA

Ejercicio 1)

Maspi, Gian Agustín

LU: 266/19

a)

Voy a eliminar los símbolos inútiles con los algoritmos vistos en la práctica. Recordemos que para esto hay que respetar el siguiente orden:

1) Eliminar inactivos
2) Eliminar inalcanzables.

Empecemos con 1)

Paso	N
0	ϕ
1	$\{A\}$
2	$\{A, E\}$
3	$\{A, E\}$

Luego de eliminar inactivos, la gramática me queda:

$$E \to A \mid \neg E \mid E \wedge E \mid (E)$$
$$A \to P$$

Luego, elimino inalcanzables, es decir, 2)

Paso	N
0	$\{E\}$
1	$\{E, A\}$
2	$\{E, A\}$

La gramática me queda de la misma forma, es decir, no hay inalcanzables

$$\boxed{\begin{array}{l} E \to A \mid \neg E \mid E \wedge E \mid (E) \\ A \to P \end{array}}$$

b) Notemos que la podemos reescribir como G_2:

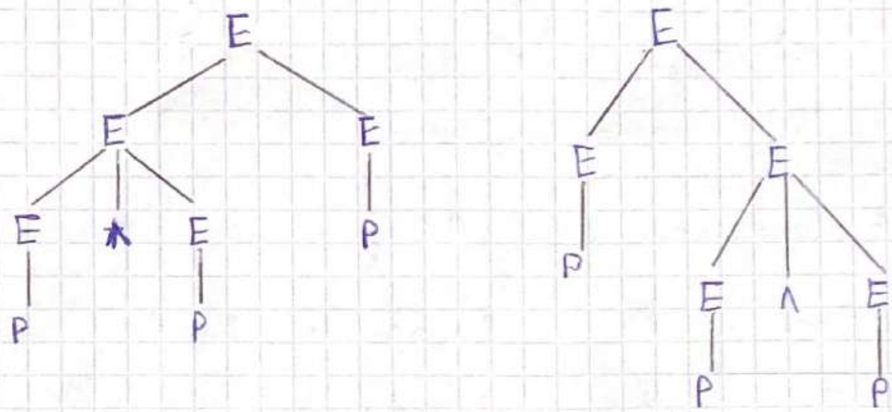
$$P \quad E \to P \mid \neg E \mid E \wedge E \mid (E)$$

$$G_2 = \langle \{E\}, \{P, \wedge, \neg, (,)\}, P, E \rangle$$

NOTA

Doy una gramática extendida que genere $L(G_1)$ y sea $ELL(1)$.

Notemos que no puedo utilizar la gramática ~~del punto~~ G_2 ~~anterior~~ ya que es ambigua. Por ej, tengo dos maneras de generar: $P \land P \land P$.



Por lo tanto, voy a dar una GLC que resuelva los conflictos de asociatividad de G_2, pero que esfuerce siempre la asociatividad a der. para no tener recursión a izq. Doy G_3' tal que

$G_3 = \langle \{E, \ , U\}, \{P, \land, \neg, (,)\}, E, P' \rangle$ con P'

$E \to U \land E \mid U$

$\cancel{T \to \land E \mid \lambda}$

$U \to \neg U \mid (E) \mid P$

El problema de esta gramática es que tiene producciones de un mismo no terminal que tienen un mismo prefijo (e): $E \to \underline{U} \land E \mid \underline{U}$. Para resolver esto factorizo a izq. Doy G_4 tal que:

$G_4 = \langle \{E, T, U\}, \{P, \land, \neg, (,)\}, E, P_1 \rangle$ con P_1

$E \to U T \qquad T \to \land E \mid \lambda \qquad U \to P \mid \neg U \mid (E)$

HOJA N°

FECHA

Veamos que G_4 es $LL(1)$

	Primeros	Siguientes
E	$\{p, \neg, (\}$	$\{), \$\}$
T	$\{\wedge\}$	$\{), \$\}$
U	$\{p, \neg, (\}$	$\{\wedge,), \$\}$

	SD
$E \to UT$	$\text{Primeros}(UT) = \{p, \neg, (\}$
$T \to \wedge E$	$\text{Primeros}(\wedge E) = \{\wedge\}$
$T \to \lambda$	$\text{Sig}(T) = \{), \$\}$
$U \to p$	$\text{Pri}(p) = \{p\}$
$U \to \neg U$	$\text{Pri}(\neg U) = \{\neg\}$
$U \to (E)$	$\text{Pri}((E)) = \{(\}$

$\cap = \phi$

$\cap = \phi$

No hay conflictos para ninguna producción. Luego, G_4 es $LL(1)$. (*)

Como, haciendo un abuso de notación, podemos interpretar a las producciones $X \to a_1 | a_2 | \ldots | a_m$ como una producción de una gramática extendida en la que $a_1 | a_2 | \ldots | a_m$ es una expresión regular, podemos decir que G_4 es una gramatica extendida. Luego, como es $LL(1)$, es $ELL(1)$

Además, ¿En la Gramatica extendida ~~vemos~~ debemos ver a $T \to \wedge E | \lambda$ como $T \to (\wedge E)$?

(*) Esto es debido a que como $\forall (A \to \beta, A \to a)$ con $a \neq \beta$ ocurre que $SD(A \to \beta) \cap SD(A \to a) = \phi$, podemos afirmar que G_4 es $LL(1)$

NOTA

c) Veamos su parser iterativo - recursivo

```
Proc E():
    U();
    T();
match("$");
accept;

Proc T():
  if (tc == "∧") {
     match("∧");
     E();
  }

Proc U():
  if (tc == "p") {
     match("p");
  } else if (tc == "¬") {
     match("¬");
     U();
  } else if (tc == "(") {
     match("(");
     E();
     match(")");
  } else error
```

Comentario ejercicio 2: Hiciste mal el análisis dado que si se pueden resolver, deberias de ver que dependiendo de cual de las opciones elijas lo que vas a lograr es tener asociatividad a izquierda o a derecha. (25/33)

FECHA

• Ej 2)

Maspi, Gian Agustín
LU: 266/19

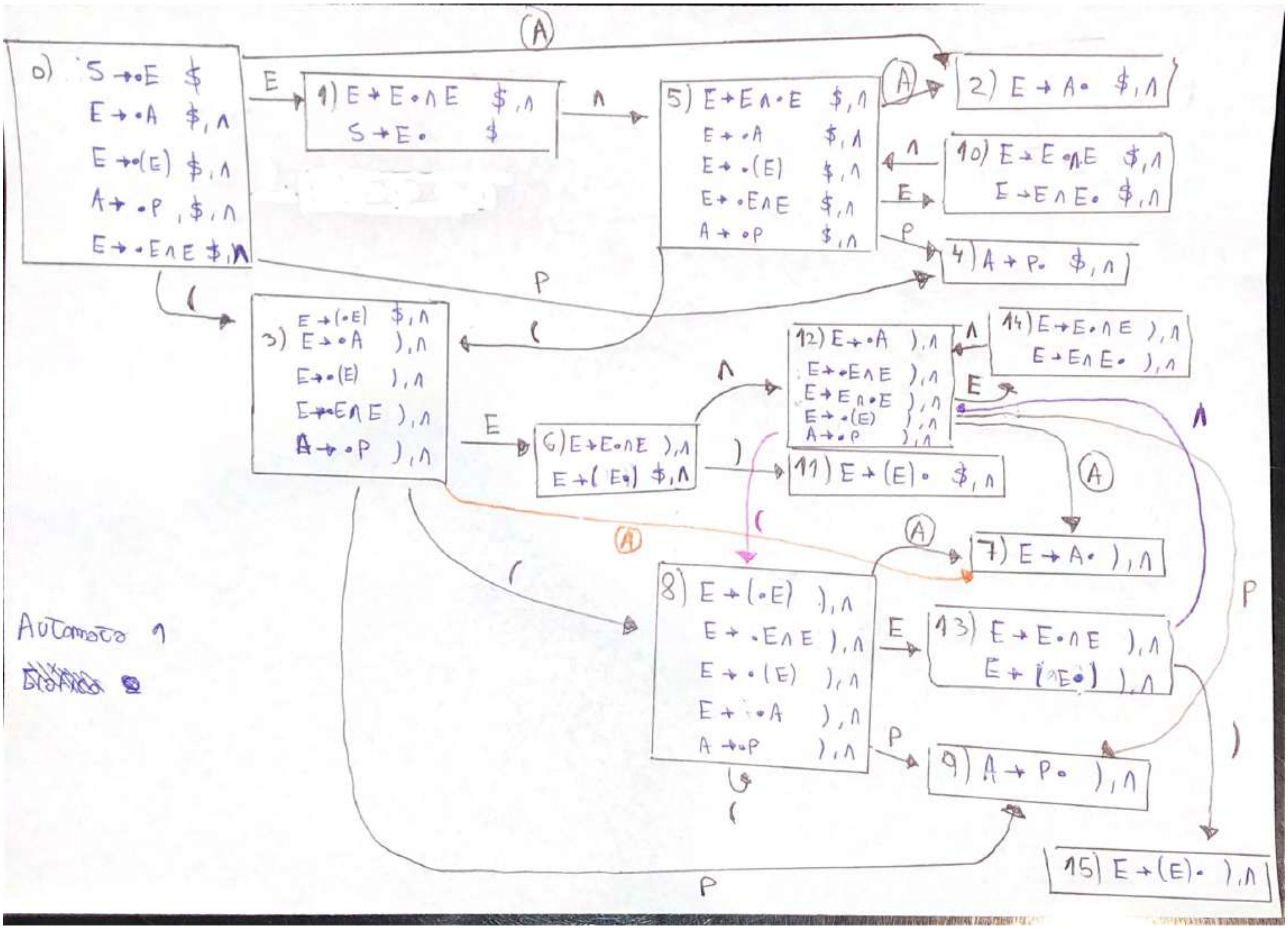
• a) ~~[illegible]~~ Automata 1

Doy la tabla LR(1)

Estado	(	)	∧	P	$	E	A
0	S3			S4		1	2
1			S5		accept		
2			r(E→A)		r(E→A)		
3	S8			S9		6	7
4			r(A→P)		r(A→P)		
5	S3			S4		10	2
6		S11	S12				
7		r(E→A)	r(E→A)				
8	S8			S9		13	7
9		r(A→P)	r(A→P)				
10			S5 r(E→E∧E)		r(E→E∧E)		
11			r(E→(E))		r(E→(E))		
12	S8			S9		14	7
13		S15	S12				
14		r(E→E∧E)	S12 r(E→E∧E)				
15		r(E→(E))	r(E→(E))				

Tenemos dos conflictos. Veamos si los podemos resolver de forma que se respete $L(E_2)$

NOTA


0) S → •E $
E → •A $, ∧
E → •(E) $, ∧
A → •P , $, ∧
E → •E∧E $, ∧
1) E → E•∧E $, ∧
S → E• $
2) E → A• $, ∧
3) E → (•E) $, ∧
E → •A), ∧
E → •(E)), ∧
E → •E∧E), ∧
A → •P), ∧
4) A → P• $, ∧
5) E → E∧•E $, ∧
E → •A $, ∧
E → •(E) $, ∧
E → •E∧E $, ∧
A → •P $, ∧
6) E → E•∧E), ∧
E → (E•) $, ∧
7) E → A•), ∧
8) E → (•E)), ∧
E → •E∧E), ∧
E → •(E)), ∧
E → •A), ∧
A → •P), ∧
9) A → P•), ∧
10) E → E•∧E $, ∧
E → E∧E• $, ∧
11) E → (E)• $, ∧
12) E → •A), ∧
E → •E∧E), ∧
E → E∧•E), ∧
E → •(E)), ∧
A → •P), ∧
13) E → E•∧E), ∧
E → (E•)), ∧
14) E → E•∧E), ∧
E → E∧E•), ∧
15) E → (E)•), ∧
A
E
∧
P
(
)

Autômato 1

HOJA N°

FECHA

Veamos que sucede si quiero parsear la cadena P ∧ P ∧ P. Hago el seguimiento:

Pila	Entrada	~~Cadena~~ Acción
0	P ∧ P ∧ P $	S4
04	∧ P ∧ P $	r(A → P)
02	∧ P ∧ P $	r(E → A)
01	∧ P ∧ P $	S5
015	P ∧ P $	S4
0154	∧ P $	r(A → P)
0152	∧ P $	r(E → A)
01510	∧ P $	1 / 2

Veamos, si elijo quedarme con 1, es decir, tomar la decisión S5 ocurre que:

Pila	Entrada	Acción
01510	∧ P $	S5
015105	P $	S4
0151054	$	r(A → P)
0151052	$	r(E → A)
01510510	$	r(E → E ∧ E)
015	$	Error

NOTA

Si nos quedamos con el shift el parser se rompe ya que no reconoce la cadena $P \wedge P \wedge P$.

Veamos que pasa si nos quedamos con el reduce, es decir, con (2)

Pila	Entrada	Acción
0 1 5 10	$\wedge$ P \$	r(E → E $\wedge$ E)
0 1	$\wedge$ P \$	S5
0 1 5	P \$	S4
0 1 5 4	\$	r(A → P)
0 1 5 2	\$	r(E → A)
0 1 5 10	\$	r(E → E $\wedge$ E)
0	\$	Error

Tomando cualquiera de las dos decisiones el parser se rompe ya que no puedo reconocer a la cadena $P \wedge P \wedge P$.
Por lo tanto, no hay forma de resolver los conflictos y que se preserve $L(G_2)$

NOTA

HOJA N°

FECHA

• Ejercicio 2)

Maspi, Gian Agustín
LU: 266/19

• b) Veamos si G_2 es LR(0)

1) Dibuje el automata.

Ahora hago la tabla para LR(0)

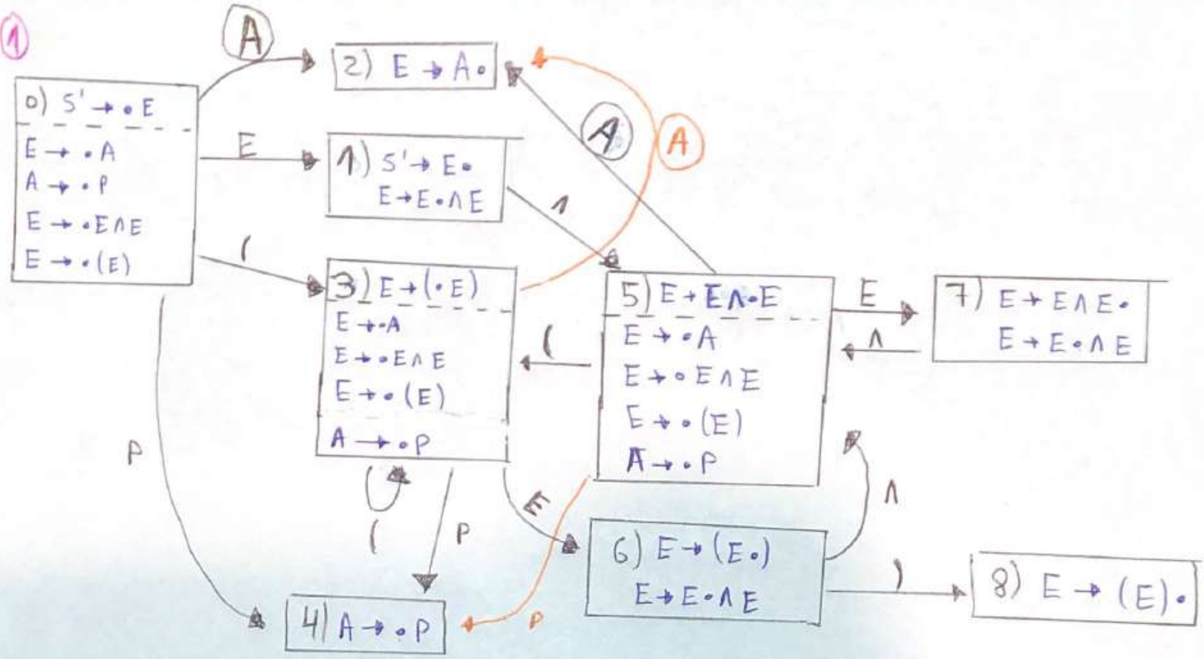
Estados	P	∧	(	)	$	E	A
0	S4		S3			1	2
1		S5			aceptar		
2	r(E→A)	r(E→A)	r(E→A)	r(E→A)	r(E→A)		
3	S4		S3			6	2
4	r(A→P)	r(A→P)	r(A→P)	r(A→P)	r(A→P)		
5	S4		S3			7	2
6		S5		S8			
7	r(E→E∧E)	S5 r(E→E∧E)	r(E→E∧E)	r(E→E∧E)	r(E→E∧E)		
8	r(E→(E))	r(E→(E))	r(E→(E))	r(E→(E))	r(E→(E))		

Tengo un conflicto para el estado 7 simbolo ∧.
shift - reduce.

Por lo tanto, G_2 NO es LR(0)

NOTA

1



• Aclaración: Redondee las A para que se diferencie del símbolo de la conjunción (∧)

Comentario ejercicio 3: Todo ok. (33/33)

HOJA N°

FECHA

Maspi, Gian Agustín
LU: 266/19

Ejercicio 3)

Atributos	Tipo	S/H
S.exp	String	Sintetizado
B.exp	String	Sintetizado
N.exp	String	Sintetizado

Reglas

$S \to S_1\ (B)$ { S.exp = S_1.exp ++ "(" ++ B.exp ++ ")" }

$S \to (N]$ { S.exp = "(" ++ N.exp ++ ")" }

$S \to \lambda$ { S.exp = "" }

$B \to B_1\ (B_2)$ { B.exp = B_1.exp ++ "(" ++ B_2.exp ++ ")" }

$B \to \lambda$ { B.exp = "" }

$N \to (B)\ N_1$ { N.exp = "(" ++ B.exp ++ ")" ++ N_1.exp }

$N \to (\ N_1$ { N.exp = "(" ++ N_1.exp ++ ")" }

$N \to \lambda$ { N.exp = "" }

NOTA